Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (original) A virtual real time system for simulating a physical test environment comprising:

a master computer module; and

at least one slave computer module communicated to the master computer module and having a clocked operation, which is synchronized to the master computer module; wherein the master computer module and at least one each slave computer module each have a launcher submodule and a deployment submodule, the launcher submodule for launching the deployment submodule and controlling the deployment submodule for synchronized operation with the master computer module, the deployment submodule generating a virtual clock and following commands from the launcher submodule.

2. (currently amended) The virtual real time system of claim 1 embodied in a workstation running under an operating system and where the launcher submodule in the master computer module is a scalable central virtual real time controller for the virtual real time system dependent on the operating system and the workstation hardware.

3. (original) The virtual real time system of claim 2 where the deployment submodule in the master computer module generates a virtual clock signal based on process CPU instruction execution.

4. (cancelled)

5. (currently amended) The virtual real time system of claim 1 further comprising a test master computer submodule communicating with the master launcher submodule for configuring the system and advancing, starting, stopping, adjusting and monitoring virtual real time, and/~~or~~ issuing time related commands to the deployment submodule in the master computer module.

6. (currently amended) The virtual real time system of claim 5 where the master deployment submodule generates a virtual clock signal and where test master computer submodule generates scale-up and/or scale-down commands of the virtual clock in the master deployment submodule during operation of the master deployment submodule.

7. (currently amended) The virtual real time system of claim 1 where the slave launcher submodule further comprises a slave launcher synch submodule and where the slave launcher submodule, upon receiving a command from ~~form~~ the master launcher submodule, requests the corresponding slave deployment submodule via the slave launcher synch submodule to advance the slave deployment submodule by a predetermined number of virtual clock ticks and to stop, after which the slave

deployment submodule suspends operation and waits for the slave launcher submodule to resume operation.

8. (currently amended) The virtual real time system of claim 1 where master launcher submodule sends a start-tick command ~~to only~~ to the slave launcher submodule, only if it is prepared to receive the next start-tick command by sending a socket call with a start-tick message.

9. (cancelled)

10. (original) The virtual real time system of claim 7 where the slave launcher submodule after receiving a start-tick command from the master deployment submodule sends a SIGCONT signal to the suspended slave deployment submodule, the slave launcher submodule sends an acknowledgment message to the master launcher submodule, the slave deployment submodule in parallel with other programs runs the requested number of ticks.

11. (original) The virtual real time system of claim 7 where the master launcher submodule then sends a signal SIGCONT to its corresponding master deployment submodule to run a requested number of virtual clock ticks based on Vclk clock ticks which are generated when the time consumed by execution of process CPU instructions is equal to or greater than tick-resolution time, the master deployment submodule

suspends its operation after running the requested number and the master launcher submodule waits for the master deployment submodule to complete its cycles.

12. (original) The virtual real time system of claim 1 where the master launcher submodule sends a stop-tick message to each slave launcher submodule which needs to be synchronized at that clock tick based on slave tick synchronize size and a stop-tick socket call is made to the candidate slave launcher submodule.

13. (original) The virtual real time system of claim 12 where the slave launcher submodule after receiving a stop-tick command waits for a SIGSTOP signal from the slave deployment submodule to make sure that the requested number of virtual clock ticks has been completed, and the slave launcher submodule sends a stop-tick acknowledgment message to the master launcher submodule.

14. (currently amended) A method for operating a virtual real time system for simulating a physical test environment comprising:

communicating at least one slave computer module with a master computer module, which at least one slave computer module has a clocked operation and is synchronized to the master computer module, wherein the master computer module and at least one each slave computer module each have a launcher submodule and a deployment submodule;

launching each of the deployment submodules corresponding to the launcher submodules;

controlling each of the deployment submodules by the corresponding launcher submodule for synchronized operation with the master computer module;~~,~~

generating a virtual clock in the deployment submodule corresponding master computer module; and

executing commands from the corresponding launcher submodule.

15. (currently amended) The method of claim 14 further comprising providing a central virtual real time controller for the virtual real time system in the launcher submodule in the master computer module, the virtual real time controller being embodied in a workstation running under an operating system, and scaling the operation of the central virtual real time controller based on the operating system and the workstation hardwire.

16. (original) The method of claim 15 where generating a virtual clock comprises generating a virtual clock signal based on process CPU instruction execution in the deployment submodule in the master computer module.

17. (cancelled)

18. (currently amended) The method of claim 14 further comprising communicating with the master launcher submodule with a test master computer submodule for configuring the system and advancing, starting, stopping, adjusting and monitoring

virtual real time, and/or issuing time related commands to the deployment submodule in the master computer module during operation of the master deployment submodule.

19. (currently amended) The method of claim 18 where generating a virtual clock in the deployment submodule corresponding master computer module comprises generating scale-up and/~~or~~ scale-down commands of the virtual clock in the master deployment submodule by means of the test master computer submodule.

20. (currently amended) The method of claim 14 where the slave launcher submodule further comprises a slave launcher synch submodule and where the slave launcher submodule, upon receiving a command from ~~form~~ the master launcher submodule, further comprising requesting the corresponding slave deployment submodule via the slave launcher synch submodule to advance the slave deployment submodule by a predetermined number of virtual clock ticks and to stop, after which the slave deployment submodule suspends operation and waits for the slave launcher submodule to resume operation.

21. (original) The method of claim 14 further comprising sending a start-tick command to only to the slave launcher submodule from the master launcher submodule, if the slave launcher submodule is prepared to receive the next start-tick command by sending a socket call with a start-tick message.

22. (cancelled)

23. (original) The method of claim 20 further comprising sending a SIGCONT signal to the suspended slave deployment submodule from the slave launcher submodule after receiving a start-tick command from the master deployment submodule, sending an acknowledgment message from the slave launcher submodule to the master launcher submodule, and running the slave deployment submodule in parallel with other programs the requested number of ticks.

24. (original) The method of claim 20 further comprising sending a signal SIGCONT from the master launcher submodule to its corresponding master deployment submodule to run a requested number of virtual clock ticks based on Vclk clock ticks which are generated when the time consumed by execution of process CPU instructions is equal to or greater than tick-resolution time, suspending operation of the master deployment submodule after running the requested number, and forcing the master launcher submodule to wait for the master deployment submodule to complete its cycles.

25. (original) The method of claim 14 further comprising sending a stop-tick message from the master launcher submodule to each slave launcher submodule which needs to be synchronized at that clock tick based on slave tick synchronize size and a stop-tick socket call is made to the candidate slave launcher submodule.

26. (original) The method of claim 25 further comprising forcing the slave launcher submodule to wait after receiving a stop-tick command for a SIGSTOP signal from the slave deployment submodule to make sure that the requested number of virtual clock ticks has been completed, and sending a stop-tick acknowledgment message from the slave launcher submodule to the master launcher submodule.